Stochastic Equilibrium Propagation for Spiking Convergent Recurrent Neural Networks

Anonymous authors
Paper under double-blind review

Abstract

Spiking Neural Networks (SNNs) promise energy-efficient, sparse, biologically inspired computation. Training them with Backpropagation Through Time (BPTT) and surrogate gradients achieves strong performance but remains biologically implausible. Equilibrium Propagation (EP) provides a more local and biologically grounded alternative. However, existing EP frameworks for SNNs largely rely on deterministic neurons, which require complex mechanisms to handle spiking discontinuities and do not scale beyond simple benchmarks such as MNIST and Fashion-MNIST. Inspired by the stochastic nature of biological spiking mechanism and recent hardware trends, we propose a stochastic EP framework that integrates probabilistic spiking neurons into the EP paradigm. This formulation smoothens the optimization landscape, stabilizes training, and enables efficient and scalable learning in SNNs. We provide theoretical guarantees showing that the proposed stochastic EP dynamics approximate deterministic EP under mean-field theory, thereby inheriting its underlying theoretical guarantees. The proposed framework narrows the performance gap with both BPTT-trained SNNs and EP-trained non-spiking convergent recurrent neural networks (CRNNs) on CIFAR-10, DVS Gesture, and IMDB datasets, while preserving temporal and spatial locality. Our results highlight stochastic EP as a promising approach for neuromorphic and on-chip learning.

1 Introduction

Spiking Neural Networks (SNNs) communicate through event-driven spikes, offering substantial energy savings from sparse activity and closer alignment with biological behaviors (Mead, 1990; Merolla et al., 2014; Benjamin et al., 2014). However, most state-of-the-art training methods still rely on Backpropagation Through Time (BPTT) with surrogate gradients (Neftci et al., 2019; He et al., 2020; Yin et al., 2021; Bal & Sengupta, 2024b; Fang et al., 2021; Zhou et al., 2023). In particular, the use of separate computational circuits for forward and backward passes is considered biologically implausible (Crick, 1989). This motivates Equilibrium Propagation (EP) (Scellier & Bengio, 2017) as a biologically grounded alternative (Figure 1). EP operates a unified circuit that relaxes neuronal states to low-energy fixed points across two phases. Small nudges at output layer implicitly propagate error information backward through feedback connections in convergent recurrent neural networks (CRNNs) with bidirectional connectivity. Synaptic weights are updated from the contrast between pre-perturbation and post-perturbation states in a Spike Timing Dependent Plasticity (STDP)-like, spatially and temporally local manner. These properties make EP particularly attractive for neuromorphic and on-chip implementations (Bi & Poo, 1998; Martin et al., 2021; Ji & Gross, 2020).

However, extending EP to the spiking domain introduces unique challenges. The binary activations and discontinuous spiking dynamics of SNNs pose challenges for deploying EP, which fundamentally relies on smooth membrane potentials for stable optimization. To address this, predictive coding mechanisms and step-size scheduling have been employed to align discrete SNN dynamics with their continuous counterparts at equilibrium (O'Connor et al., 2019; Lin et al., 2024). A relaxed, lightweight Leaky Integrate-and-Fire (LIF) neuron formulation has also been explored within EP (Martin et al., 2021). Yet, scalability remains limited, and accuracy degrades as layers deepen, with evaluations largely restricted to simple visual benchmarks. This

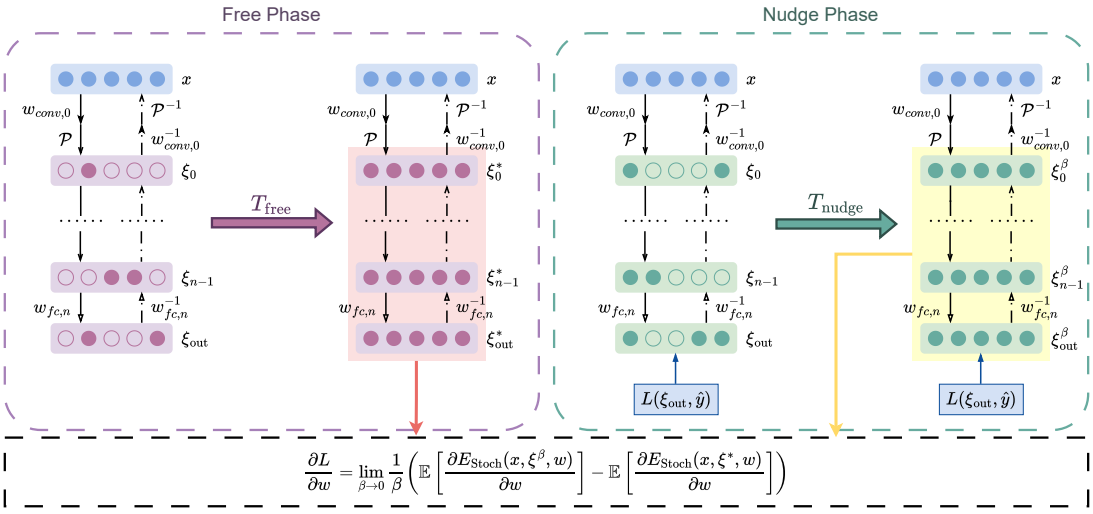

Figure 1: Equilibrium Propagation (EP) optimizes neural networks through two phases. With inputs x clamped, the network state ξ relaxes to a fixed point ξ^* after T_{free} time steps. A weak teaching signal then nudges only the output units ξ_{out} toward their target labels $\hat{y}$ by adding a small perturbation term to the dynamics, producing a nearby fixed point ξ^β for a small $\beta > 0$ after T_{nudge} time steps. Each synapse updates based on the contrast between its equilibrium states across the two phases. Circles with outlines represent unsaturated neurons, while filled circles denote saturated ones. Purple and green correspond to the free and nudge phases, respectively.

highlights an open challenge to extend EP to richer architectures and modern datasets while maintaining stability without introducing additional complexity.

Furthermore, current energy-based SNN frameworks (Martin et al., 2021; O'Connor et al., 2019; Lin et al., 2024) primarily employ deterministic neurons (Lapicque, 1907) with rate-based coding schemes. However, these formulations complicate optimization due to discontinuities arising from spike generation, refractory membrane potentials, and membrane potential resetting mechanisms (Izhikevich, 2003). In contrast, stochastic spiking modulation (Jiang et al., 2024), inspired by the biological observation that neurons fire probabilistically (Maass, 2015), remains largely underexplored in energy-based SNNs. Within the context of energy-based models, particularly under EP, optimization efficiency heavily depends on the smoothness of membrane potentials. Stochastic SNNs address this challenge by preserving smooth membrane dynamics and alleviating discontinuities compared to deterministic neurons. Moreover, optimizing over continuously differentiable likelihoods of the spike generation process (Jiang et al., 2024; Bagheri et al., 2018) makes direct training of stochastic models practical in both BPTT and EP frameworks.

Motivated by these observations, we develop a stochastic EP framework for training spiking networks. We provide a principled bridge between stochastic spiking networks and EP by *deriving an EP-compatible objective via mean-field analysis*. Rather than relying on an ad-hoc stabilization, we *verify the mean-field conditions* and relate the stochastic spiking dynamics to an equivalent rate-model objective, yielding an *energy-consistent, theoretically grounded* learning update. The contributions of this work are:

1. To the best of our knowledge, this is the first framework that trains stochastic SNNs within the EP paradigm, extending EP beyond deterministic LIF formulations.

2. The proposed framework achieves stable training and scales to deep convolutional spiking CRNNs, addressing the depth and instability challenges observed in prior EP-trained SNNs.

3. We provide theoretical guarantees showing that the proposed framework is equivalent to the deterministic EP formulation under mean-field theory, ensuring convergence and approximating BPTT.

4. The framework achieves state-of-the-art performance among EP-trained SNNs and BPTT-trained SNNs and narrows the gap to EP-trained non-spiking counterparts, while maintaining reduced computational cost.

2 Related Work

2.1 Stochastic SNNs

While most studies on SNNs rely on deterministic neuron models, substantial evidence indicates that biological neurons fire stochastically (Faisal et al., 2008; Maass, 2015). Inspired by this observation, stochastic SNNs incorporate probabilistic mechanisms into spike generation or synaptic transmission, offering richer representational capacity and improved robustness. Early works established the theoretical link between stochastic spiking dynamics and probabilistic inference. For example, Buesing et al. (2011) showed that recurrent SNNs implement Markov Chain Monte Carlo sampling, while Pecevski et al. (2011) demonstrated approximate inference in graphical models. Ma et al. (2023) further explored noisy SNNs that leverage intrinsic neural noise as a computational resource for scalable and reliable computation.

More recently, stochastic SNNs have been extended to large-scale learning tasks. Jiang et al. (2024) employed stochastic LIF neurons with First-To-Spike coding to improve robustness and reduce latency, whereas Yao et al. (2025) utilized a stochastic Spike Response Model with expectation propagation to enable efficient probabilistic training for supervised tasks. Additionally, Bal & Sengupta (2024a) proposed a probabilistic spiking state-space model that leverages stochastic spike generation to efficiently capture long-range dependencies. Meanwhile, emerging device trends in neuromorphic hardware have further fueled interest in stochastic SNNs (Sengupta et al., 2016; Yang & Sengupta, 2020). State-compressed SNNs operating in the probability domain (mapped to scaled stochastic devices) achieve accuracy comparable to multi-bit deterministic counterparts (Islam et al., 2024; 2023), while stochasticity itself enhances fault tolerance (Ardakani et al., 2021) and mitigates overfitting (Hinton et al., 2012b). Collectively, these studies underscore the promise of stochastic spiking neurons in addressing diverse computational tasks. Consequently, coupling the benefits of stochasticity with EP represents a compelling direction for advancing neuromorphic computing and enabling on-chip learning.

2.2 Local Learning Algorithms

Although SNNs offer substantial algorithmic and hardware advantages, most state-of-the-art training approaches still rely on BPTT with surrogate gradients (Neftci et al., 2019). These methods remain biologically implausible due to their dependence on weight symmetry (Lillicrap et al., 2016; Nøkland, 2016; Frenkel et al., 2021), global error propagation (Baldi et al., 2017), and explicit gradient computations (Crick, 1989; Lillicrap et al., 2020). To address these limitations, several local learning methods have been proposed (Lin et al., 2025b; Lin & Sengupta, 2025). Feedback Alignment (FA) replaces symmetric weights with fixed random matrices (Lillicrap et al., 2016), while Direct Feedback Alignment (DFA) transmits errors directly from the output layer to hidden layers (Nøkland, 2016; Bellec et al., 2020). Local Error (LE) extends this concept by assigning each layer its own cost function and propagating pseudo-targets through random mappings (Kaiser et al., 2020; Frenkel et al., 2021). EP provides another biologically grounded and theoretically equivalent alternative to BPTT (Scellier & Bengio, 2017). Analyses show that EP closely approximates BPTT (Scellier & Bengio, 2017; 2019). However, early EP variants suffered from first-order bias under finite nudging, restricting architectures to shallow fully connected networks (Scellier & Bengio, 2017; O'Connor et al., 2019; Ernoult et al., 2019; 2020). Three-phase training procedures have been proposed to mitigate this bias (Laborieux et al., 2021) and reduce the performance gap with BPTT-based baselines on both vision and sequential NLP tasks (Lin et al., 2025a; Bal & Sengupta, 2023). Collectively, these approaches advance biological plausibility and computational efficiency, making them promising candidates for neuromorphic and on-chip learning.

2.3 EP in Spiking CRNN Architectures

EP has also been explored in spiking CRNNs. Prior work has primarily relied on deterministic spiking neurons, particularly LIF models (Martin et al., 2021; O'Connor et al., 2019; Lin et al., 2024). A central

challenge in these approaches lies in smoothing membrane potentials to maintain stable optimization within the EP framework. For instance, Martin et al. (2021) introduced a low-pass moving average filter for LIF neurons, but this method was restricted to shallow linear layers, where deeper stacking led to significant accuracy degradation. Alternatively, predictive encoding and decoding strategies with step-size scheduling have been proposed to better align discrete-time SNN dynamics with the fixed points of non-spiking CRNNs (O'Connor et al., 2019; Lin et al., 2024). Despite these efforts, EP-trained SNNs have struggled to scale beyond shallow architectures and simple visual benchmarks, with accuracy degrading as network depth increases. This persistent limitation underscores the need for approaches that stabilize spiking dynamics while preserving EP's spatial and temporal locality. To the best of our knowledge, the proposed framework is the first to integrate stochastic neurons directly into EP, combining the smooth membrane potential dynamics of stochastic SNNs with the locality advantages of energy-based learning.

3 Methods

3.1 Stochastic Spiking Neuron

Inspired by prior work (Jiang et al., 2024; Bagheri et al., 2018; Laydevant et al., 2021), we design stochastic spiking neurons (Figure 2) that are compatible with the proposed stochastic energy function. The membrane potential ξ^t is formulated as a linear combination of the previous membrane potential and the input contributions across the temporal dimension (the formula will be derived later in Equation 7). Subsequently, a hard sigmoid function is applied to compute the firing probability at time t, and spikes $s^t \in \{0, 1\}$ are generated through Bernoulli sampling:

$$s^t = \mathcal{B}(\sigma(\xi^t)) \tag{1}$$

$$\sigma(x) = \begin{cases} 0 & \text{if } x \leq 0 \\ 1 & \text{if } x \geq 1 \\ \kappa x & \text{otherwise} \end{cases} \tag{2}$$

where κ is a tunable scaling factor that controls the firing frequency. Although spike generation is inherently non-differentiable, this issue is mitigated in our proposed EP framework because gradients are computed from local changes in the membrane potential ξ^t, which is a continuous variable. In contrast to deterministic LIF neurons, stochastic spiking neurons naturally exhibit smoother dynamics (Section 4.1), as they avoid hard resets and refractory periods that cause abrupt state changes and complicate optimization within the EP framework.

3.2 Stochastic Energy Function and Neuron Dynamics

We now formalize the proposed stochastic EP framework as an underlying energy-based model. It builds on a continuous-valued Hopfield network with symmetric recurrent connections, characterized by the Hopfield energy function (Hopfield, 1984; Scellier & Bengio, 2017). Extending this formulation to layered architectures (Laydevant et al., 2021; Ernoult et al., 2019; Scellier & Bengio, 2019) yields the following energy expression (we omit the bias term and input layer for simplicity):

$$E(x, \xi^t, w) = \tfrac{1}{2} \sum_{i=0}^{N_\mathrm{t}-1} \left\| \xi_i^t \right\|^2 - \sum_{i=0}^{N_\mathrm{t}-2} \rho(\xi_i^t)^\mathsf{T} w_i \rho(\xi_{i+1}^t) \tag{3}$$

Here, ξ_i^t denotes the neuron states at layer i, w_i represents the synaptic weight, $\rho(\cdot)$ is the activation function, and N_t denotes the total number of layers. The corresponding dynamics are given by

$$\frac{\partial \xi^t}{\partial t} = -\frac{\partial E(x, \xi^t, w)}{\partial \xi} \tag{4}$$

To incorporate stochastic spiking behavior into the computational circuit of EP, which provides high sparsity and significantly reduces computational cost (Section 5.1), we extend Equation 3 into a stochastic version

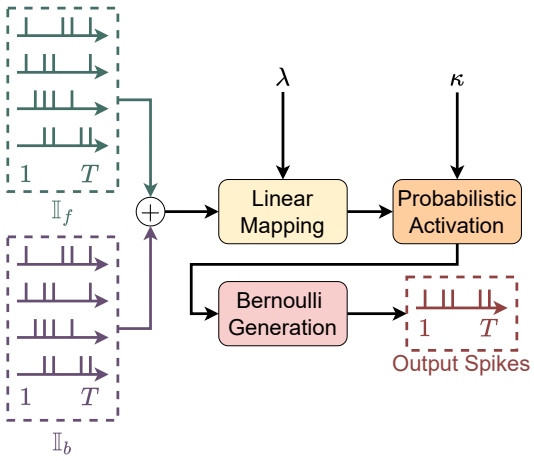

Figure 2: Illustration of stochastic spiking neuron dynamics. The membrane potential integrates weighted inputs from both forward $\mathbb{I}_f$ and backward passes $\mathbb{I}_b$ with decay factor λ, which is mapped to a firing probability scaled by factor κ, and generates spikes through Bernoulli sampling.

of the energy function by replacing the deterministic activation function with a stochastic spike generation process:

$$E_{\text{Stoch}}(x, \xi^t, w) = \tfrac{1}{2} \sum_{i=0}^{N_t-1} \|\xi_i^t\|^2 - \sum_{i=0}^{N_t-2} \mathcal{B}(\sigma(\xi_i^t))^\mathsf{T} w_i \mathcal{B}(\sigma(\xi_{i+1}^t)) \tag{5}$$

Here, $\mathcal{B}(\cdot)$ denotes Bernoulli sampling and $\sigma(\cdot)$ is a probabilistic activation function. To address the non-differentiability of Bernoulli sampling, the straight-through (ST) estimator (Hinton et al., 2012a; Bengio et al., 2013) for $\mathcal{B}'(\cdot)$ is employed, thereby yielding valid gradients for stochastic neurons. Derivation of neuron dynamics based on Equation 5 is defined as:

$$\begin{aligned} \frac{\partial \mathcal{B}(\sigma(\xi^t))}{\partial \xi^t} &= \mathcal{B}'(\sigma(\xi^t))\sigma'(\xi^t) \approx 1 \cdot \sigma'(\xi^t) \\ \frac{\partial \xi_i^t}{\partial t} &= \sigma'(\xi_i^t)\Big(w_i \mathcal{B}(\sigma(\xi_{i+1}^t)) + w_{i-1}^\mathsf{T} \mathcal{B}(\sigma(\xi_{i-1}^t))\Big) - \xi_i^t \end{aligned} \tag{6}$$

Based on the Euler method with step size λ (interpretable as the decay rate of the membrane potential), the discretized neuron dynamics can be expressed as:

$$\xi_i^{t+1} = (1-\lambda)\xi_i^t + \lambda \sigma'(\xi_i^t)\Big(w_i \mathcal{B}(\sigma(\xi_{i+1}^t)) + w_{i-1}^\mathsf{T} \mathcal{B}(\sigma(\xi_{i-1}^t))\Big) \tag{7}$$

With the straight-through estimator approximation, Equation 7 preserves the stochastic spiking generation process in inter-neuron communication through $\mathcal{B}(\sigma(\cdot))$, while the gradient computation of the activation depends solely on the continuous function $\sigma(\cdot)$. The convergence of the proposed neuron dynamics is demonstrated in Section 4.1.

3.3 Equilibrium Propagation

The training procedure of EP is adapted to perform gradient descent in spiking CRNNs on a loss function $L(\xi_{\text{out}}^t, \hat{y})$ defined between the target $\hat{y}$ and the output activations ξ_{out}^t (Figure 1) (Ernoult et al., 2019; O'Connor et al., 2019; Lin et al., 2024; Martin et al., 2021). In the first (free) phase, a static input x is presented, and the network evolves over T_{free} time steps until it converges to a stable state ξ^* that minimizes the stochastic energy function $E_{\text{Stoch}}(x, \xi^t, w)$. In the second (nudge) phase, a nudging term $\beta \frac{\partial L(\xi_{\text{out}}^t, \hat{y})}{\partial \xi}$ with β denoting the nudging factor, is applied to the output layer ξ_{out}^t. The resulting neuron dynamics are given

by

$$\frac{\partial \xi^t}{\partial t} = -\frac{\partial E_{\text{Stoch}}(x, \xi^t, w)}{\partial \xi} - \beta \frac{\partial L(\xi^t_{\text{out}}, \hat{y})}{\partial \xi} \tag{8}$$

This nudging term shifts the saturated states of the network toward another stable state ξ^β, which lies closer to the true label after T_{nudge} time steps. Then the weight gradients can be computed by

$$\frac{\partial L}{\partial w} = \lim_{\beta \to 0} \frac{1}{\beta} \left(\frac{\partial E_{\text{Stoch}}(x, \xi^\beta, w)}{\partial w} - \frac{\partial E_{\text{Stoch}}(x, \xi^*, w)}{\partial w} \right) \tag{9}$$

However, accurate estimation of weight gradients (Scellier & Bengio, 2017; Ernoult et al., 2019) depends on stable neuron states, whereas Equation 9 yields a noisy optimization landscape due to fluctuations introduced by the stochastic spike generation process. In extreme cases, $\sum_{i=0}^{N_t-2} \mathcal{B}(\sigma(\xi_i^t))^\mathsf{T} w_i \mathcal{B}(\sigma(\xi_{i+1}^t))$ may vanish when no spikes are generated at time t. To address this issue, we establish the equivalence between the proposed stochastic energy function and its deterministic counterpart under mean-field theory (see Appendix H for the formal proof), as stated in the following theorem.

Theorem 3.1. *Suppose the energy function (Equation 3 and Equation 5) has symmetric weights. The mean firing rate satisfies:*

$$\mathbb{E}[\mathcal{B}(\sigma(\xi_i^t))] = \sigma(\xi_i^t) \tag{10}$$

Under a mean-field independence assumption for sufficiently large networks (Sompolinsky et al., 1988; Buice et al., 2010), we assume the units are independent across neuronal indices ($j \neq k$).

$$\mathbb{E}[\mathcal{B}(\sigma(\xi_j^t))\mathcal{B}(\sigma(\xi_k^t))] = \sigma(\xi_j^t)\sigma(\xi_k^t) \tag{11}$$

Then, given that neuron states ξ are deterministic, and $\sigma(\cdot) = \rho(\cdot)$, the expected stochastic energy (Equation 13) equals the deterministic energy function.

$$\mathbb{E}[E_{\text{Stoch}}(x, \xi^t, w)] = E(x, \xi^t, w) \tag{12}$$

where

$$\mathbb{E}[E_{\text{Stoch}}(x, \xi^t, w)] = \tfrac{1}{2} \sum_{i=0}^{N_t-1} \|\xi_i^t\|^2 - \sum_{i=0}^{N_t-2} \sigma(\xi_i^t)^\mathsf{T} w_i \sigma(\xi_{i+1}^t) \tag{13}$$

Leveraging Theorem 3.1 mitigates the instability caused by directly computing weight gradients from E_{Stoch}. It is worth mentioning here that the independence of neuronal activities is a common assumption in neural network models (La Camera, 2021). We make two key observations:

1. The expected stochastic energy coincides with the deterministic energy under mean-field theory, effectively constructing a deterministic rate model.
2. The neuron dynamics (Equation 7 and Equation 8) resemble a moving average of firing rates, where the saturated states of stochastic neurons in the free and nudge phases represent their mean firing rates. This correspondence yields stable states for gradient computation.

Based on these observations, we estimate the weight gradients in stochastic EP using the expected energy function, which approximates the gradients of a rate-based deterministic EP and therefore preserves its theoretical foundations, as stated in the following theorem.

Theorem 3.2. *The gradient of the objective function L with respect to w can be estimated from the divergence of the two stable states:*

$$\frac{\partial L}{\partial w} = \lim_{\beta \to 0} \frac{1}{\beta} \left(\mathbb{E}\left[\frac{\partial E_{\text{Stoch}}(x, \xi^\beta, w)}{\partial w} \right] - \mathbb{E}\left[\frac{\partial E_{\text{Stoch}}(x, \xi^*, w)}{\partial w} \right] \right) \tag{14}$$

We refer the reader to Scellier & Bengio (2019) for the theoretical justification of Theorem 3.2 in deterministic settings, and to Ernoult et al. (2019) and Laborieux et al. (2021) for explicit formulations of the input functions in convolutional and linear layers.

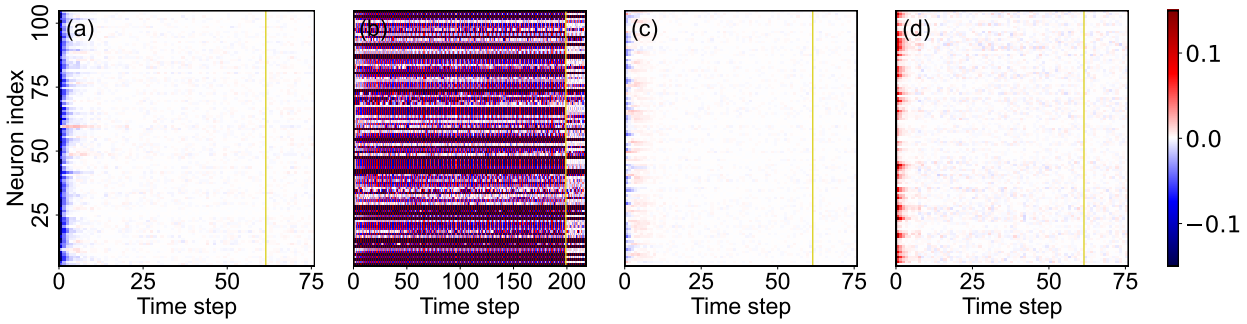

Figure 3: Comparison of membrane potential stability across different spiking neuron models. Heatmaps show changes in membrane potential over time for a network with one hidden layer of 512 neurons trained on 100 random MNIST samples. (a) The proposed stochastic model exhibits stable dynamics, rapidly converging to a smooth equilibrium. (b) A deterministic LIF model with a low-pass filter (Martin et al., 2021) shows pronounced instability. (c) A deterministic LIF model stabilized using predictive coding and step-size scheduling (O'Connor et al., 2019; Lin et al., 2024) achieves convergence. (d) A deterministic LIF model with only step-size scheduling, simplified from (O'Connor et al., 2019; Lin et al., 2024), exhibits fluctuations. The yellow vertical dashed line indicates the transition from the free phase to the nudge phase.

3.4 Augmenting the Number of Output Neurons

Laydevant et al. (2021) argues that, in EP frameworks with binary activations, effective backward error transmission to upstream layers requires a sufficient number of output neurons whose activation function changes satisfy:

$$\Delta\xi = |\xi^{\beta} - \xi^{*}| > \frac{1}{2} \tag{15}$$

To achieve this, we follow prior work by increasing the error signal through output-layer augmentation, where each prediction neuron is replaced by N_{perclass} neurons per class, inflating the output layer from N_{classes} to $N_{\text{classes}} \times N_{\text{perclass}}$.

4 Experiments

4.1 Stability of Stochastic Spiking CRNN

In this section, we conduct a toy experiment using a single hidden layer of 512 neurons trained on the MNIST dataset to analyze membrane potential stability across different spiking neuron models. A single-layer configuration is chosen due to the scalability limitations of the deterministic baseline (Martin et al., 2021). The objective is to show that stochastic spiking neurons trained with the proposed stochastic EP framework converge to stable states and exhibit higher stability than deterministic LIF neurons. All models are trained to reach at least 75% accuracy to ensure convergence, and membrane potentials are averaged over 100 randomly selected MNIST samples.

Figure 3 compares the stability of the proposed stochastic model against deterministic baselines. The heatmaps track the temporal evolution of membrane potentials as the network transitions from the free phase to the nudge phase. Our stochastic model consistently converges to an equilibrium with minimal potential variance across neurons, whereas deterministic methods encounter notable stability issues. The LIF model with a low-pass filter (Martin et al., 2021) shows strong instability. Incorporating predictive coding and step-size scheduling (O'Connor et al., 2019; Lin et al., 2024) improves stability in deterministic models but increases computational cost (Section 5.1). Removing predictive coding reveals that step-size scheduling alone cannot maintain stable membrane dynamics. These findings demonstrate that the stochastic formulation provides a more efficient and stable solution for EP training in SNNs.

Table 1: Testing accuracy (%) of spiking and non-spiking CRNNs trained using the EP framework on MNIST, CIFAR-10, DVS Gesture, and IMDB datasets. Results are reported as the mean over five independent runs, with standard deviation shown in parentheses. Here, #FC denotes the number of hidden fully connected layers (excluding output layers) and #C the number of convolutional layers. #HA refers to the number of Hopfield attention layers. *Bin.* (Y/N) indicates whether the model employs binary activations, *Optim.* denotes the optimization method, and *Local* (Y/N) specifies whether the optimization approach is a local learning method.

Model	**Bin.**	**Optim.**	**Local**	**Accuracy**
1FC(O'Connor et al., 2019)	Y	EP	Y	97.63
1FC(Martin et al., 2021)	Y	EP	Y	97.59
2FC(Laydevant et al., 2021)	Y	EP	Y	96.97
2C(Ernoult et al., 2019)	N	EP	Y	98.98
2C1FC(Lin et al., 2024)	Y	EP	Y	99.03
2C1FC(Lin et al., 2024)	Y	BPTT	N	99.14
2C1FC(Lin et al., 2024)	N	BPTT	N	99.01
1FC (Ours)	Y	EP	Y	97.46
2FC (Ours)	Y	EP	Y	98.05
2C (Ours)	Y	EP	Y	98.99

(a) Evaluated on the MNIST dataset.

Model	**Bin.**	**Optim.**	**Local**	**Accuracy**
VGG-5(Laborieux et al., 2021)	N	EP	Y	87.55
VGG-5(Laydevant et al., 2021)	N	EP	Y	84.34
VGG-5(Laborieux & Zenke, 2022)	N	BPTT	N	88.30
5C (Massa et al., 2020)	Y	BPTT	N	78.92
ReActNet (Chen et al., 2021)	Y	BPTT	N	77.60
5C (Lillicrap et al., 2014)	Y	FA	N	71.48 (0.42)
5C (Xiao et al., 2021)	Y	ID	Y	76.25 (0.39)
5C (Ours)	Y	EP	Y	81.33 (0.18)
5CP (Ours)	Y	EP	Y	81.73 (0.19)

(b) Evaluated on the CIFAR-10 dataset.

Model	**Bin.**	**Optim.**	**Local**	**Accuracy**
2C1FC(Viale et al., 2021)	Y	BPTT	N	78.70
2C1FC(Stewart et al., 2020)	Y	SOEL	Y	64.70
2C1FC(Stewart et al., 2020)	Y	SLAYER(Shrestha & Orchard, 2018)	N	83.50
3C (Ours)	Y	EP	Y	89.92 (0.74)

(c) Evaluated on the DVS Gesture dataset.

Model	**Bin.**	**Optim.**	**Local**	**Accuracy**
1HA2FC(Bal & Sengupta, 2022)	N	EP	Y	88.9
1HA1FC (Ours)	Y	EP	Y	80.99 (0.30)

(d) Evaluated on the IMDB dataset.

Table 2: GPU memory consumption (in MB) of CRNNs trained using the EP framework on the MNIST, CIFAR-10, DVS Gesture, and IMDB datasets. #FC represents the number of hidden linear layers, and #C denotes the number of convolutional layers. #HA refers to the number of Hopfield attention layers. The batch size is set to 128 for the MNIST, CIFAR-10, and DVS Gesture datasets, and to 64 for the IMDB dataset. The notation $N\times$ indicates that BPTT consumes N times more GPU memory than EP.

Dataset	MNIST			CIFAR-10	DVS Gesture	IMDB
Model	1FC	2FC	2C	5C	3C	1HA1FC
EP	141	147	281	7885	4617	12267
BPTT	179 (1.2×)	195 (1.3×)	677 (2.4×)	42938 (5.4×)	10381 (2.2×)	18189 (1.5×)

4.2 Dataset and Network Topology

The proposed framework is evaluated on standard datasets, including MNIST, CIFAR-10, DVS Gesture, and IMDB datasets (details in Appendix C). The architectural configurations for each dataset are provided in Appendix D. Model-specific hyperparameters were optimized individually for optimal performance and summarized in Appendix E. All experiments were implemented in Python using the PyTorch framework and executed on an NVIDIA RTX A5000 GPU with 24 GB of memory. The resulting performance is reported in Table 1.

4.3 Performance

The proposed stochastic EP framework substantially narrows the performance gap between non-spiking models and their BPTT-trained counterparts (Table 1). To the best of our knowledge, this work presents the first evaluation of EP-trained SNNs on complex datasets such as CIFAR-10, DVS Gesture (see Appendix A for additional details), and IMDB datasets. **Notably, the framework achieves comparable performance compared to iso-architecture SNNs trained with BPTT.** Although its accuracy on CIFAR-10 and IMDB remains approximately 3% to 8% lower than that of full-precision models, it offers significantly reduced computational and memory costs (see Section 4.4 and Section 5.1). We further evaluate an alternative stochastic mechanism for SNNs based on a Poisson escape-rate process (Amarasingham et al., 2006; Buxó & Pillow, 2020; Ding et al., 2025), which yields performance comparable to Bernoulli spike generation (denoted 5CP in Table 1), demonstrating the generalizability of the proposed formulation. Compared to other local learning approaches, including feedback alignment (FA) (Lillicrap et al., 2014) and implicit differentiation (ID) (Xiao et al., 2021), our framework achieves superior performance on CIFAR-10. (Note that the results for FA and ID are produced using our re-implementation of the original methods in order to ensure a consistent architecture across all comparisons.) Overall, these results highlight the effectiveness, generalizability, and scalability of the proposed stochastic EP framework.

4.4 Memory Consumption

The GPU memory consumption of spiking CRNNs trained with EP and BPTT is reported in Table 2. All experiments were conducted with a batch size of 128 using the architectures described in Section 4.2. The number of time steps for each model was set according to the optimal hyperparameters yielding the best performance, as detailed in Appendix E. The results demonstrate that the EP framework offers a more memory efficient training scheme for deep SNNs. EP achieves substantial memory savings while maintaining performance comparable to BPTT-trained counterparts. Notably, these memory savings remain significant even when the dimensionality of the output layers is inflated.

5 Ablation Studies

5.1 Computational Cost in Stochastic EP Spiking Models

We quantify computational complexity using the number of Multiply–Accumulate Operations (MACs) and accumulation operations (ACs). In this work, we estimate MACs and ACs using the same formulation for both forward and backward computational paths.

Computational cost compared to full-precision models. We estimate inference energy efficiency on neuromorphic hardware by comparing the number of spiking computations with the operation count of an iso-architecture full-precision model trained with EP (Table 2). Following standard cost formulations (Lu & Sengupta, 2020; Howard et al., 2017; Sandler et al., 2018), the number of operations for the i-th layer in an full-precision network (unidirectional) is defined as

$$\mathrm{MAC}_i^{\mathrm{full-precision}} = \begin{cases} C_{\mathrm{in}} K_H K_W C_{\mathrm{out}} O_H O_W & \text{if } 1 \le i \le N_{\mathrm{c}} \\ I_F O_F & \text{if } N_{\mathrm{c}} < i \le N_{\mathrm{t}} \end{cases} \tag{16}$$

Here, C_{in} and C_{out} are the input and output channels, $K_H \times K_W$ is the kernel size, $O_H \times O_W$ are the output dimensions, and I_F, O_F are the input and output features. N_{c} and N_{t} denote the number of convolutional and total layers, respectively. The total number of operations (unidirectional) in the full-precision model is therefore $\mathrm{MAC}_{\mathrm{tot}}^{\mathrm{full-precision}} = \sum_{i=1}^{N_{\mathrm{t}}} \mathrm{MAC}_i^{\mathrm{full-precision}}$. Unlike full-precision networks, SNNs operate on sparse, event-driven signals, where computation occurs only when spikes are generated. For a layer with an average firing rate IFR_i, the number of synaptic operations (unidirectional) in the proposed stochastic framework is $\mathrm{AC}_{\mathrm{tot}}^{\mathrm{SNN}} = \sum_{i=1}^{N_{\mathrm{t}}} IFR_i \times \mathrm{MAC}_i^{\mathrm{full-precision}}$.

We estimate the total energy consumption on the MNIST dataset using the hyperparameters in Appendix E for the 2FC and 2C architectures. For CRNNs, the computation of $\mathrm{MAC}_{\mathrm{tot}}^{\mathrm{full-precision}}$ and $\mathrm{AC}_{\mathrm{tot}}^{\mathrm{SNN}}$ is performed in both forward and backward directions. To evaluate the worst-case scenario, we assume the SNN architectures employ inflated output neurons (700 neurons at output layer), while the full-precision models use the number of classes as output neurons (10 neurons at output layer). The average firing rates IFR_i are [0.21, 0.19, 0.12] and [0.27, 0.19, 0.10] for the 2FC and 2C architectures, respectively (averaged over 128 samples). Using hardware energy costs of 0.9 pJ for ACs and 4.6 pJ for MACs (Han et al., 2015), total energy consumption is computed by multiplying the energy per operation by the corresponding operation count. Under these settings, the proposed framework achieves 19× and 23× reductions in energy consumption relative to full-precision models.

Computational cost compared to deterministic spiking model. We analyze the computational efficiency of the proposed method against deterministic SNNs trained with EP. Prior works on scalable spiking CRNNs (O'Connor et al., 2019; Lin et al., 2024) integrate predictive coding into neuron dynamics within the EP framework, formulated as

$$\begin{aligned} \boxed{Dec^t} &= (1-\alpha) \cdot Dec^{t-1} + \alpha \cdot \mathbb{I} \\ \xi^t &= (1-\lambda)\xi^{t-1} + \lambda \rho'(\xi^{t-1}) Dec^t \\ \boxed{Enc^t} &= \frac{1}{\alpha} \cdot \left[\rho(\xi^t) - (1-\alpha) \cdot \rho(\xi^{t-1})\right] \\ s^t &= \delta(V^{t-1} + Enc^t > \mathcal{V}_{th}) \\ \boxed{V^t} &= V^{t-1} + Enc^t - s^t \end{aligned} \tag{17}$$

where $\mathbb{I}$ is the weighted inputs, and $\delta(\cdot)$ governs deterministic spike generation based on the membrane potential V^t and threshold $\mathcal{V}_{th}$. The predictive decoder estimates the current signal Dec^t from previous samples with predictive factor α, and the predictive encoder Enc^t quantizes temporal changes in neuron states into binary signals. Assuming a total of I neurons and equivalent spike generation cost as in the stochastic framework (Equation 1), predictive coding adds $4I$ multiplications (red in Equation 17) and $4I$

additions (yellow in Equation 17) per time step to update Dec^t and Enc^t. Each layer further requires $3I$ additional memory units to store intermediate variables Dec^t, Enc^t, and V^t (boxed in Equation 17). These overheads highlight the efficiency of our stochastic formulation, which achieves stable dynamics without predictive coding.

5.2 Effect of Increasing Number of Output Neurons

The inflation factor N_{perclass} scales to strengthen error signals. Insufficient learning signals are observed in models with binary activations (Laydevant et al., 2021), including our stochastic spiking network, where neuron activations can be zeroed out during the backpropagation of error signals. N_{perclass} amplifies the error signal $\frac{\partial L(\xi^t_{\text{out}},\hat{y})}{\partial \xi}$ (Appendix G) to ensure it is sufficiently potent to perturb the network state from its free-phase equilibrium ξ^* towards the nudge-phase equilibrium ξ^β at nudge phase. An insufficient nudge yields $\xi^* \approx \xi^\beta$, causing weight updates to vanish. However, merely increasing N_{perclass} does not guarantee better performance. In our experiments, it is carefully tuned to prevent vanishing gradient rather than to optimize performance directly.

6 Discussion

The proposed stochastic EP framework shows that incorporating stochastic spiking neurons into Equilibrium Propagation yields stable learning dynamics and competitive performance on standard vision benchmarks. Under mean-field theory, it can be viewed as optimizing an energy landscape over neuronal firing rates, providing theoretical grounding for convergence and equivalence to BPTT. By mitigating discontinuities inherent in deterministic spiking models, stochastic spiking neurons improves training stability and smoothens optimization, narrowing the performance gap between EP and surrogate-gradient BPTT while preserving the efficiency and locality of energy-based learning. These results establish stochasticity as both a biologically plausible and computationally advantageous mechanism for neuromorphic and on-chip learning. Although the framework scales to deeper convolutional CRNNs, extending it to large-scale architectures and datasets such as ResNet (He et al., 2016) on ImageNet (Russakovsky et al., 2015) remains an open challenge. Subsequent research should also encompass hardware-level deployments to fully assess the neuromorphic potential of stochastic EP.

References

Asohan Amarasingham, Ting-Li Chen, Stuart Geman, Matthew T Harrison, and David L Sheinberg. Spike count reliability and the poisson hypothesis. *Journal of Neuroscience*, 26(3):801–809, 2006.

Arnon Amir, Brian Taba, David Berg, Timothy Melano, Jeffrey McKinstry, Carmelo Di Nolfo, Tapan Nayak, Alexander Andreopoulos, Guillaume Garreau, Marcela Mendoza, et al. A low power, fully event-based gesture recognition system. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7243–7252, 2017.

Amir Ardakani, Arash Ardakani, and Warren J Gross. Fault-tolerance of binarized and stochastic computing-based neural networks. In *2021 IEEE Workshop on Signal Processing Systems (SiPS)*, pp. 52–57. IEEE, 2021.

Alireza Bagheri, Osvaldo Simeone, and Bipin Rajendran. Training probabilistic spiking neural networks with first-to-spike decoding. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2986–2990. IEEE, 2018.

Malyaban Bal and Abhronil Sengupta. Sequence learning using equilibrium propagation. *arXiv preprint arXiv:2209.09626*, 2022.

Malyaban Bal and Abhronil Sengupta. Sequence learning using equilibrium propagation. In *2023 32nd International Joint Conference on Artificial Intelligence (IJCAI)*, 2023.

Malyaban Bal and Abhronil Sengupta. P-spikessm: Harnessing probabilistic spiking state space models for long-range dependency tasks. *arXiv preprint arXiv:2406.02923*, 2024a.

Malyaban Bal and Abhronil Sengupta. Spikingbert: Distilling bert to train spiking language models using implicit differentiation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 10998–11006, 2024b.

Pierre Baldi, Peter Sadowski, and Zhiqin Lu. Learning in the machine: the symmetries of the deep learning channel. *Neural Networks*, 95:110–133, 2017.

Guillaume Bellec, Franz Scherr, Anand Subramoney, Elias Hajek, Darjan Salaj, Robert Legenstein, and Wolfgang Maass. A solution to the learning dilemma for recurrent networks of spiking neurons. *Nature communications*, 11(1):3625, 2020.

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

Ben Varkey Benjamin, Peiran Gao, Emmett McQuinn, Swadesh Choudhary, Anand R Chandrasekaran, Jean-Marie Bussat, Rodrigo Alvarez-Icaza, John V Arthur, Paul A Merolla, and Kwabena Boahen. Neurogrid: A mixed-analog-digital multichip system for large-scale neural simulations. *Proceedings of the IEEE*, 102 (5):699–716, 2014.

Guo-qiang Bi and Mu-ming Poo. Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type. *Journal of neuroscience*, 18(24):10464–10472, 1998.

Lars Buesing, Johannes Bill, Bernhard Nessler, and Wolfgang Maass. Neural dynamics as sampling: a model for stochastic computation in recurrent networks of spiking neurons. *PLoS computational biology*, 7(11): e1002211, 2011.

Michael A Buice, Jack D Cowan, and Carson C Chow. Systematic fluctuation expansion for neural network activity equations. *Neural computation*, 22(2):377–426, 2010.

Camille E Rullán Buxó and Jonathan W Pillow. Poisson balanced spiking networks. *PLoS computational biology*, 16(11):e1008261, 2020.

Tianlong Chen, Zhenyu Zhang, Xu Ouyang, Zechun Liu, Zhiqiang Shen, and Zhangyang Wang. " bnn-bn=?": Training binary neural networks without batch normalization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4619–4629, 2021.

Francis Crick. The recent excitement about neural networks. *Nature*, 337:129–132, 1989.

Mike Davies, Narayan Srinivasa, Tsung-Han Lin, Gautham Chinya, Yongqiang Cao, Sri Harsha Choday, Georgios Dimou, Prasad Joshi, Nabil Imam, Shweta Jain, et al. Loihi: A neuromorphic manycore processor with on-chip learning. *Ieee Micro*, 38(1):82–99, 2018.

Jianhao Ding, Zhaofei Yu, Jian K Liu, and Tiejun Huang. Neuromorphic computing paradigms enhance robustness through spiking neural networks. *Nature Communications*, 16(1):10175, 2025.

Maxence Ernoult, Julie Grollier, Damien Querlioz, Yoshua Bengio, and Benjamin Scellier. Updates of equilibrium prop match gradients of backprop through time in an rnn with static input. *Advances in neural information processing systems*, 32, 2019.

Maxence Ernoult, Julie Grollier, Damien Querlioz, Yoshua Bengio, and Benjamin Scellier. Equilibrium propagation with continual weight updates, 2020.

S Esser, P Merolla, J Arthur, A Cassidy, R Appuswamy, A Andreopoulos, D Berg, J McKinstry, T Melano, D Barch, et al. Convolutional networks for fast, energy-efficient neuromorphic computing. 2016. *Preprint on ArXiv. http://arxiv. org/abs/1603.08270. Accessed*, 27, 2016.

A Aldo Faisal, Luc PJ Selen, and Daniel M Wolpert. Noise in the nervous system. *Nature reviews neuroscience*, 9(4):292–303, 2008.

Wei Fang, Zhaofei Yu, Yanqi Chen, Tiejun Huang, Timothée Masquelier, and Yonghong Tian. Deep residual learning in spiking neural networks. *Advances in Neural Information Processing Systems*, 34:21056–21069, 2021.

Charlotte Frenkel, Martin Lefebvre, and David Bol. Learning without feedback: Fixed random learning signals allow for feedforward training of deep neural networks. *Frontiers in neuroscience*, 15:629892, 2021.

Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. *Advances in neural information processing systems*, 28, 2015.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Weihua He, YuJie Wu, Lei Deng, Guoqi Li, Haoyu Wang, Yang Tian, Wei Ding, Wenhui Wang, and Yuan Xie. Comparing snns and rnns on neuromorphic vision datasets: Similarities and differences. *Neural Networks*, 132:108–120, 2020.

Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. *Cited on*, 14(8):2, 2012a.

Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012b.

John J Hopfield. Neurons with graded response have collective computational properties like those of two-state neurons. *Proceedings of the national academy of sciences*, 81(10):3088–3092, 1984.

Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

ANM Islam, Arnob Saha, Zhouhang Jiang, Kai Ni, and Abhronil Sengupta. Hybrid stochastic synapses enabled by scaled ferroelectric field-effect transistors. *Applied Physics Letters*, 122(12), 2023.

ANM Nafiul Islam, Kezhou Yang, Amit Kumar Shukla, Pravin Khanal, Bowei Zhou, Wei-Gang Wang, and Abhronil Sengupta. Hardware in loop learning with spin stochastic neurons. *Advanced Intelligent Systems*, 6(7):2300805, 2024.

Eugene M Izhikevich. Simple model of spiking neurons. *IEEE Transactions on neural networks*, 14(6): 1569–1572, 2003.

Zhengyun Ji and Warren Gross. Towards efficient on-chip learning using equilibrium propagation. In *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–5. IEEE, 2020.

Yi Jiang, Sen Lu, and Abhronil Sengupta. Stochastic spiking neural networks with first-to-spike coding. In *2024 International Conference on Neuromorphic Systems (ICONS)*, pp. 24–31. IEEE, 2024.

Jacques Kaiser, Hesham Mostafa, and Emre Neftci. Synaptic plasticity dynamics for deep continuous local learning (decolle). *Frontiers in Neuroscience*, 14:424, 2020.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images, 2009.

Giancarlo La Camera. The mean field approach for populations of spiking neurons. In *Computational modelling of the brain: Modelling approaches to cells, circuits and networks*, pp. 125–157. Springer, 2021.

Axel Laborieux and Friedemann Zenke. Holomorphic equilibrium propagation computes exact gradients through finite size oscillations. *Advances in neural information processing systems*, 35:12950–12963, 2022.

Axel Laborieux, Maxence Ernoult, Benjamin Scellier, Yoshua Bengio, Julie Grollier, and Damien Querlioz. Scaling equilibrium propagation to deep convnets by drastically reducing its gradient estimator bias. *Frontiers in neuroscience*, 15:633674, 2021.

Louis Édouard Lapicque. Louis lapicque. *J. physiol*, 9:620–635, 1907.

Jérémie Laydevant, Maxence Ernoult, Damien Querlioz, and Julie Grollier. Training dynamical binary neural networks with equilibrium propagation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4640–4649, 2021.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 2002.

Timothy P Lillicrap, Daniel Cownden, Douglas B Tweed, and Colin J Akerman. Random feedback weights support learning in deep neural networks. *arXiv preprint arXiv:1411.0247*, 2014.

Timothy P Lillicrap, Daniel Cownden, Douglas B Tweed, and Colin J Akerman. Random synaptic feedback weights support error backpropagation for deep learning. *Nature communications*, 7(1):13276, 2016.

Timothy P Lillicrap, Adam Santoro, Luke Marris, Colin J Akerman, and Geoffrey Hinton. Backpropagation and the brain. *Nature Reviews Neuroscience*, 21(6):335–346, 2020.

Jiaqi Lin and Abhronil Sengupta. Toward spiking neural network local learning modules resistant to adversarial attacks. *arXiv preprint arXiv:2504.08897*, 2025.

Jiaqi Lin, Malyaban Bal, and Abhronil Sengupta. Scaling snns trained using equilibrium propagation to convolutional architectures. In *2024 International Conference on Neuromorphic Systems (ICONS)*, pp. 312–318. IEEE, 2024.

Jiaqi Lin, Malyaban Bal, and Abhronil Sengupta. Scalable equilibrium propagation via intermediate error signals for deep convolutional crnns. *arXiv preprint arXiv:2508.15989*, 2025a.

Jiaqi Lin, Sen Lu, Malyaban Bal, and Abhronil Sengupta. Benchmarking spiking neural network learning methods with varying locality. *IEEE Access*, 2025b.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Sen Lu and Abhronil Sengupta. Exploring the connection between binary and spiking neural networks. *Frontiers in neuroscience*, 14:535, 2020.

Gehua Ma, Rui Yan, and Huajin Tang. Exploiting noise as a resource for computation and learning in spiking neural networks. *Patterns*, 4(10), 2023.

Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pp. 142–150, 2011.

Wolfgang Maass. To spike or not to spike: that is the question. *Proceedings of the IEEE*, 103(12):2219–2224, 2015.

Erwann Martin, Maxence Ernoult, Jérémie Laydevant, Shuai Li, Damien Querlioz, Teodora Petrisor, and Julie Grollier. Eqspike: spike-driven equilibrium propagation for neuromorphic implementations. *Iscience*, 24(3), 2021.

Riccardo Massa, Alberto Marchisio, Maurizio Martina, and Muhammad Shafique. An efficient spiking neural network for recognizing gestures with a dvs camera on the loihi neuromorphic processor. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–9. IEEE, 2020.

Carver Mead. Neuromorphic electronic systems. *Proceedings of the IEEE*, 78(10):1629–1636, 1990.

Paul A Merolla, John V Arthur, Rodrigo Alvarez-Icaza, Andrew S Cassidy, Jun Sawada, Filipp Akopyan, Bryan L Jackson, Nabil Imam, Chen Guo, Yutaka Nakamura, et al. A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science*, 345(6197):668–673, 2014.

Emre O Neftci, Hesham Mostafa, and Friedemann Zenke. Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks. *IEEE Signal Processing Magazine*, 36(6):51–63, 2019.

Arild Nøkland. Direct feedback alignment provides learning in deep neural networks. *Advances in neural information processing systems*, 29, 2016.

Peter O'Connor, Efstratios Gavves, and Max Welling. Training a spiking neural network with equilibrium propagation. In *The 22nd international conference on artificial intelligence and statistics*, pp. 1516–1523. PMLR, 2019.

Dejan Pecevski, Lars Buesing, and Wolfgang Maass. Probabilistic inference in general graphical models through sampling in stochastic networks of spiking neurons. *PLoS computational biology*, 7(12):e1002294, 2011.

Christian Pehle, Sebastian Billaudelle, Benjamin Cramer, Jakob Kaiser, Korbinian Schreiber, Yannik Stradmann, Johannes Weis, Aron Leibfried, Eric Müller, and Johannes Schemmel. The brainscales-2 accelerated neuromorphic system with hybrid plasticity. *Frontiers in Neuroscience*, 16:795876, 2022.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543, 2014.

Hubert Ramsauer, Bernhard Schäfl, Johannes Lehner, Philipp Seidl, Michael Widrich, Thomas Adler, Lukas Gruber, Markus Holzleitner, Milena Pavlović, Geir Kjetil Sandve, et al. Hopfield networks is all you need. *arXiv preprint arXiv:2008.02217*, 2020.

Kaushik Roy, Akhilesh Jaiswal, and Priyadarshini Panda. Towards spike-based machine intelligence with neuromorphic computing. *Nature*, 575(7784):607–617, 2019.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510–4520, 2018.

Benjamin Scellier and Yoshua Bengio. Equilibrium propagation: Bridging the gap between energy-based models and backpropagation. *Frontiers in computational neuroscience*, 11:24, 2017.

Benjamin Scellier and Yoshua Bengio. Equivalence of equilibrium propagation and recurrent backpropagation. *Neural computation*, 31(2):312–329, 2019.

Abhronil Sengupta, Maryam Parsa, Bing Han, and Kaushik Roy. Probabilistic deep spiking neural systems enabled by magnetic tunnel junction. *IEEE Transactions on Electron Devices*, 63(7):2963–2970, 2016.

Sumit B Shrestha and Garrick Orchard. Slayer: Spike layer error reassignment in time. *Advances in neural information processing systems*, 31, 2018.

Haim Sompolinsky, Andrea Crisanti, and Hans-Jurgen Sommers. Chaos in random neural networks. *Physical review letters*, 61(3):259, 1988.

Kenneth Stewart, Garrick Orchard, Sumit Bam Shrestha, and Emre Neftci. Online few-shot gesture learning on a neuromorphic processor. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 10(4):512–521, 2020.

Alberto Viale, Alberto Marchisio, Maurizio Martina, Guido Masera, and Muhammad Shafique. Carsnn: An efficient spiking neural network for event-based autonomous cars on the loihi neuromorphic research processor. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–10. IEEE, 2021.

Mingqing Xiao, Qingyan Meng, Zongpeng Zhang, Yisen Wang, and Zhouchen Lin. Training feedback spiking neural networks by implicit differentiation on the equilibrium state. *Advances in neural information processing systems*, 34:14516–14528, 2021.

Kezhou Yang and Abhronil Sengupta. Stochastic magnetoelectric neuron for temporal information encoding. *Applied Physics Letters*, 116(4), 2020.

Dan Yao, Steve McLaughlin, and Yoann Altmann. Training of spiking neural networks with expectation-propagation. *arXiv preprint arXiv:2506.23757*, 2025.

Bojian Yin, Federico Corradi, and Sander M Bohté. Accurate and efficient time-domain classification with adaptive spiking recurrent neural networks. *Nature Machine Intelligence*, 3(10):905–913, 2021.

Friedemann Zenke and Surya Ganguli. Superspike: Supervised learning in multilayer spiking neural networks. *Neural computation*, 30(6):1514–1541, 2018.

Zhaokun Zhou, Yuesheng Zhu, Chao He, Yaowei Wang, Shuicheng YAN, Yonghong Tian, and Li Yuan. Spikformer: When spiking neural network meets transformer. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=frE4fUwz_h`.

A Evaluation on Neuromorphic Datasets

Unlike static datasets, samples in the DVS Gesture dataset (Amir et al., 2017) are time-varying event sequences, whereas the standard EP framework is formulated for static inputs. Directly applying EP to temporal data leads to non-convergent energy dynamics because the saturated neuron states vary from frame to frame. To address this, we adapt the framework by introducing separate training and evaluation loops. For each temporal step τ in an input sequence, both the free and nudge phases are executed for T_{free} and T_{nudge} iterations, respectively, allowing the network to reach equilibrium at every τ. Weight updates are then computed using the saturated states $\xi^{\tau,*}$ and $\xi^{\tau,\beta}$ at each time step. The total runtime per sample becomes $T_{\text{total}} = \tau_{\text{total}} \times (T_{\text{free}} + T_{\text{nudge}})$, where τ_{total} denotes the total number of time steps in each input sequence. Final predictions are obtained by summing the output activations $\xi_{\text{out}}^{\tau,*}$ across all τ.

Additionally, we propose a warm-up initialization that relates the initial state for frame at time $\tau+1$ to the saturated state for frame at time τ. Concretely, instead of resetting to zero, we initialize as a linear function of the previous equilibrium:

$$\xi^{\tau+1,0} = \mathbf{0} + \gamma \xi^{\tau,*}, \tag{18}$$

where $\mathbf{0}$ is the zero state and $\gamma \in [0, 1]$ controls the degree of temporal state retention. This design is motivated by the following assumptions:

1. **Temporal similarity:** consecutive frames in DVS Gesture exhibit high similarity due to the underlying event-based dynamics.
2. **Stability of equilibrium:** if two static input frames are similar, the network dynamics converge to similar equilibrium ($\xi^{\tau+1,*} \approx \xi^{\tau,*}$ when inputs are similar).

B Hopfield Attention in EP Framework

While the above procedure enables EP to operate on event streams, many sequential domains, such as language tasks, are more naturally modeled by explicit sequence mechanisms rather than frame-wise equilibration alone. A widely used approach is to incorporate attention-based modules to capture long-range dependencies. In the EP literature, Hopfield attention has been explored in full-precision settings (Bal & Sengupta, 2022). Here, we adapt and integrate Hopfield attention into our proposed EP framework. Specifically, the state transition dynamics are governed by the Hopfield attention module, formulated as:

$$\frac{\partial \xi^t}{\partial t} = \text{HopAttn}\left(-\frac{\partial E_{\text{Stoch}}(x, \xi^t, w)}{\partial \xi}, K\right) \tag{19}$$

The HopAttn operator yields an attention-based embedding of the input sequence, defined by scaled dot-product attention:

$$\text{HopAttn}(Q, K) = \text{softmax}\left(\frac{1}{\sqrt{d_k}} Q K^{\mathsf{T}}\right) K \tag{20}$$

Here, d_k denotes the encoding dimension of the key K. Q is the query and can be interpreted as the current state pattern. In our experiments, we apply Hopfield attention only at the first layer. Specifically, Q is formed by adding the initial input x, projected into the query space via W_Q, to the feedback signals from the next layer. The key K consists of the sequence of stored patterns in the Hopfield network, typically corresponding to the network input x. Crucially, the Hopfield attention module carries both theoretical and empirical guarantees of converging to a steady state after a single update step (Ramsauer et al., 2020; Bal & Sengupta, 2022). By leveraging this property, the system is driven toward stable states at every time step.

C Datasets

We evaluate on three standard vision benchmarks spanning increasing complexity and sensing modalities: MNIST (static grayscale digits), CIFAR-10 (natural RGB images), and DVS Gesture (event-based neuromorphic recordings). These datasets enable a balanced evaluation across both static and spiking regimes.

For the DVS Gesture dataset, we use 60 time steps per sample. We additionally evaluate our framework on IMDB as a language benchmark. Word representations for the IMDB dataset are initialized using 300-dimensional GloVe embeddings (Pennington et al., 2014). These embeddings serve as input to the model, with input sequences truncated or padded to a maximum length of 256 tokens.

MNIST. The MNIST dataset contains 60,000 training and 10,000 test images of handwritten digits (LeCun et al., 2002).

CIFAR-10. CIFAR-10 comprises 32×32 color images across ten object categories, posing a more challenging classification task than MNIST (Krizhevsky et al., 2009).

DVS Gesture. The DVS Gesture dataset includes recordings from 29 subjects captured under three distinct lighting conditions, providing a neuromorphic benchmark with event-driven dynamics (Amir et al., 2017).

IMDB. The IMDB dataset consists of 50,000 highly polar movie reviews split evenly into 25,000 training and 25,000 test samples, serving as a standard benchmark for binary sentiment classification (Maas et al., 2011).

D Network Topologies

In this section, we detail the architectural topology used for each dataset. For MNIST, we evaluate three architectures. Two fully connected models (1FC and 2FC) with one and two hidden layers of 512 neurons each, and a 2C model composed of two convolutional layers with 64 and 128 channels, followed by a linear layer mapping features to the output dimension. Each convolutional layer uses a kernel size of 5, stride 1, and padding 1, followed by a 3×3 max pooling operation with stride 3. For CIFAR-10, we adopt a 5C architecture consisting of four convolutional layers with 64, 128, 256, and 256 channels, and a linear layer mapping features to the output. Each layer employs a kernel size of 5, stride 2, and padding 2, followed by a 2×2 max pooling operation with stride 2. In the 5CP setting, spike generation follows a Poisson escape-rate process. The probability of emitting a spike at time t is given by

$$P(s^t = 1 \mid \xi^t) = 1 - \exp\left(-\sigma(\xi^t)\Delta t\right), \tag{21}$$

where $\Delta t = 1$ denotes the simulation time step and $\kappa = 1$ is the activation scaling factor. For the DVS Gesture dataset, we use a 3C model comprising three convolutional layers with 64, 128, and 256 channels, and a linear layer mapping features to the output. Each convolutional layer applies a kernel size of 5, stride 1, and padding 1, followed by a 2×2 max pooling operation with stride 2. For the IMDB dataset, we employ a single attention module at the first layer (following (Bal & Sengupta, 2022)), followed by a linear readout layer that maps the extracted features to the output logits.

E Hyperparameters

This section outlines the hyperparameter configurations utilized in the experiments. Table 3 summarizes the hyperparameters used for the MNIST, CIFAR-10, DVS Gesture, and IMDB datasets. For MNIST, the network is optimized using stochastic gradient descent (SGD) without momentum. For CIFAR-10, DVS Gesture, and IMDB datasets, the AdamW optimizer (Loshchilov & Hutter, 2017) is employed with a weight decay of 0.001, 0.0001, and 0.01 respectively.

Additionally, conventional EP (Scellier & Bengio, 2017) introduces an estimation bias due to the positive nudging factor β, leading to inaccurate gradient estimates. Laborieux et al. (2021) addressed this issue by introducing a randomized β and a three-phase training procedure. The additional phase applies a nudging factor of $-\beta$ to counteract the bias. In this work, we adopt the same three-phase training strategy. Let $\xi^{-\beta}$ denote the steady state of the network at the end of the third phase. Under this formulation, Equation 14 becomes

$$\frac{\partial L}{\partial w} = \lim_{\beta \to 0} \frac{1}{\beta} \left(\mathbb{E}\left[\frac{\partial E_{\text{Stoch}}(x, \xi^{\beta}, w)}{\partial w} \right] - \mathbb{E}\left[\frac{\partial E_{\text{Stoch}}(x, \xi^{-\beta}, w)}{\partial w} \right] \right) \tag{22}$$

which preserves both spatial and temporal locality in updating the synaptic weights between connected layers. The use of randomized β and three-phase training is summarized in Table 3.

Table 3: Hyper-parameters for optimal performance on various datasets. LR represents the learning rates and BS represents the batch sizes.

Model	λ	T_{free}	T_{nudge}	β	κ	N_{perclass}	**LR**	**BS**	**Epoch**	**Bias Issue**	γ
MNIST											
1FC	0.5	60	15	0.75	2	10	3e-3	4	100	Random	-
2FC	0.5	60	15	0.5	2	70	2e-2	64	200	Random	-
2C	0.5	150	50	0.5	2	70	5e-4	16	200	Random	-
CIFAR-10											
5C	0.3	250	50	0.15	1	50	1e-4	100	200	3-Phase	-
DVS Gesture											
3C	0.25	150	35	0.25	1	50	1e-4	64	100	3-Phase	0.3
IMDB											
Attention	0.5	150	50	0.5	1	100	1e-5	25	200	3-Phase	-

F Scaling Factor, Firing Density, and Sparsity

Prior to Bernoulli sampling, activations are scaled by a gain κ. Larger κ increases firing density and correspondingly decreases sparsity. This value provides an effective balance between ensuring sufficient spike activity for information propagation and maintaining a high degree of sparsity, which is a key property that enables SNNs to achieve energy-efficient computation (Roy et al., 2019; Zenke & Ganguli, 2018; Esser et al., 2016; Davies et al., 2018; Pehle et al., 2022).

In addition, we analyze how the scaling factor κ (Equation 1) influences network sparsity and the noise level. If firing becomes too low, error signals fail to propagate and learning stalls. In this regime, we increase κ to raise spike rates. Conversely, if firing becomes too high, the phase contrast that drives EP weight updates is diminished. We then decrease κ to restore a useful contrastive signal. To visualize sparsity in SNNs, we varied κ to control the spike firing frequency, which is sampled from a Bernoulli distribution based on the membrane potentials. The results are demonstrated in Figure 4.

G Effects of Output Layer Inflation Factor

In this section, we investigate the effect of the output-layer inflation factor N_{perclass} on the strength of error signals. We quantify how increasing the number of output neurons influences gradient propagation during training. The experiment is conducted using the 5C architecture on the CIFAR-10 dataset, where the error signals are evaluated by averaging over 128 random samples (Appendix 5). As N_{perclass} increases, a consistent rise in the summed error-signal magnitude is observed, indicating that output-layer inflation effectively amplifies gradient flow and enhances feedback propagation in the stochastic EP framework. Importantly, the percentage of non-zero error signals remains largely unchanged across different values of the inflation factor N_{perclass} (Appendix 6). However, since the total number of output neurons increases with larger N_{perclass}, the overall magnitude of received error signals also grows accordingly.

H Proof of Equivalence for Stochastic EP Framework and Deterministic EP Framework under Mean-field Theory

In this section, we provide the formal proof of Theorem 3.1.

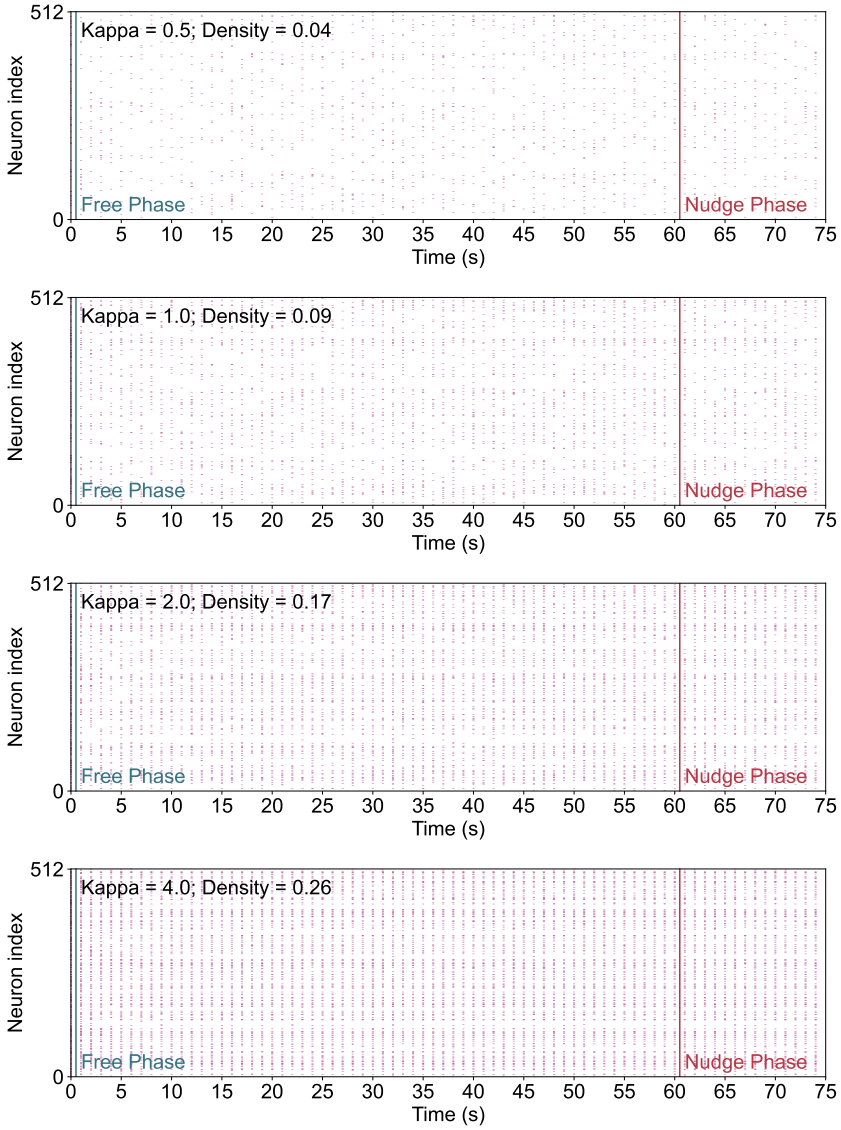

Figure 4: Spike raster plots of hidden-layer neurons trained on MNIST for different scaling factors κ. As κ increases from 0.5 to 4.0, the firing density rises from 0.04 to 0.26, indicating more frequent spiking activity. Each plot shows neuron firing patterns during the free and nudge phases.

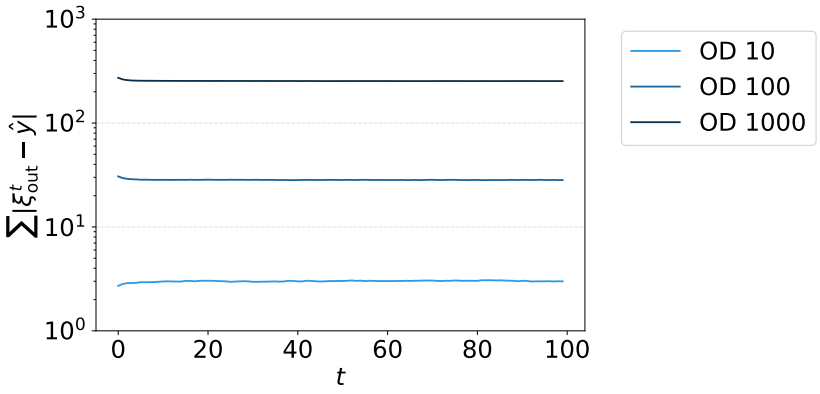

Figure 5: Summed magnitude of error signals at the output layer, averaged over 128 samples from the CIFAR-10 dataset using the 5C architecture. OD denotes the total number of output neurons (10, 100, and 1,000) corresponding to $N_{\text{perclass}} = 1, 10$, and 100, respectively. Increasing N_{perclass} amplifies the output-layer error signals, confirming stronger gradient propagation during training.

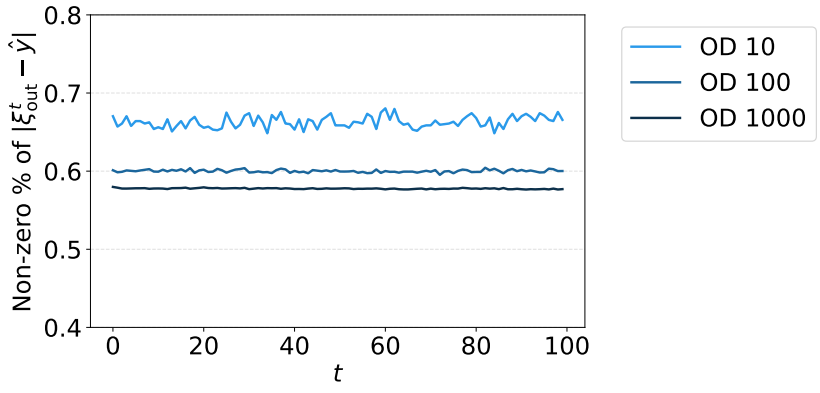

Figure 6: Percentage of non-zero error signals at the output layer, averaged over 128 samples from the CIFAR-10 dataset using the 5C architecture. OD denotes the total number of output neurons (10, 100, and 1,000), corresponding to $N_{\text{perclass}} = 1, 10$, and 100, respectively.

Proof. Assume throughout that (i) $\sigma(\cdot) = \rho(\cdot)$, where ρ is the activation function used in the deterministic model, and (ii) conditioned on the membrane states ξ, the output spikes are independent Bernoulli random variables. By definition in Equation 5, the stochastic energy function is defined as:

$$E_{\text{Stoch}}(x, \xi^t, w) = \tfrac{1}{2} \sum_{i=0}^{N_t-1} \|\xi_i^t\|^2 - \sum_{i=0}^{N_t-2} \mathcal{B}(\sigma(\xi_i^t))^\mathsf{T} w_i \mathcal{B}(\sigma(\xi_{i+1}^t)) \tag{23}$$

Taking expectation of $E_{\text{Stoch}}(x, \xi^t, w)$:

$$\mathbb{E}\left[E_{\text{Stoch}}(x, \xi^t, w)\right] = \tfrac{1}{2} \sum_{i=0}^{N_t-1} \|\xi_i^t\|^2 - \sum_{i=0}^{N_t-2} \mathbb{E}\left[\mathcal{B}(\sigma(\xi_i^t))^\mathsf{T} w_i \mathcal{B}(\sigma(\xi_{i+1}^t))\right] \tag{24}$$

By the mean-field theory for Bernoulli distribution $\mathbb{E}[\mathcal{B}(\sigma(\xi^t))] = \sigma(\xi^t) = \rho(\xi^t)$ and independence across neurons,

$$\mathbb{E}\left[\mathcal{B}(\sigma(\xi_i^t))\right] = \rho(\xi_i^t) \tag{25}$$

$$\mathbb{E}\left[\mathcal{B}(\sigma(\xi_j^t))\mathcal{B}(\sigma(\xi_k^t))\right] = \rho(\xi_j^t)\rho(\xi_k^t) \quad (j \neq k) \tag{26}$$

$$\mathbb{E}\left[\mathcal{B}(\sigma(\xi_i^t))^\mathsf{T} w_i \mathcal{B}(\sigma(\xi_{i+1}^t))\right] = \rho(\xi_i^t)^\mathsf{T} w_i \rho(\xi_{i+1}^t) \tag{27}$$

Hence, by the linearity of expectation, we have:

$$\mathbb{E}\left[E_{\text{Stoch}}(x, \xi^t, w)\right] = \tfrac{1}{2} \sum_{i=0}^{N_t-1} \|\xi_i^t\|^2 - \sum_{i=0}^{N_t-2} \rho(\xi_i^t)^\mathsf{T} w_i \rho(\xi_{i+1}^t) = E(x, \xi^t, w) \tag{28}$$

which gives $\mathbb{E}[E_{\text{stoch}}(x, \xi^t, w)] = E(x, \xi^t, w)$. Since $E(x, \xi^t, w)$ is established as a Lyapunov function for the deterministic state dynamics (Scellier & Bengio, 2017), the expected stochastic energy serves as a Lyapunov function in expectation and is monotonically decreasing along trajectories, implying convergence of the neuron states. □

Note that this convergence analysis is not tied to a specific noise distribution. As long as stochastic spike generation can be estimated by its expected firing rate, this proof can be extended with minor changes by replacing sampled spikes with their mean-field expectations. Thus, the same convergence argument applies.